Іваницька Анна, Лабораторна №8, Звіт

Під час виконання програми створюються публічний і приватний ключі та зберігаються в контейнері та файлі відповідно.

Код:

using System;

using System.Text;

using System.Security.Cryptography;

using System.ComponentModel;

namespace lab8

{

class Program

{

static void Main(string[] args)

{

// initializing string option, data

string option, data;

// paths

string path = "D:/Lessons/2nd Year/Основи інформаційної безпеки/Labs/Lab 8/lab8",

publicKeyPath = path + "/publicKey.xml",

privateKeyPath = path + "/privateKey.xml";

// initializing array

byte[] hashedDocument;

do

{

Console.WriteLine();

Console.WriteLine(" | Options:");

Console.WriteLine(" | 1 - digital signature");

Console.WriteLine(" | 0 - exit");

Console.WriteLine();

Console.Write(" Your option -> ");

option = Console.ReadLine();

Console.WriteLine();

Console.WriteLine();

//

if (option == "1")

{

var cspParams = new DigitalSignature();

Console.Write(" Enter your data -> ");

data = Console.ReadLine(); // getting data

Console.WriteLine();

Console.WriteLine();

var document = Encoding.UTF8.GetBytes(data); // string data to bytes

using (var sha256 = SHA256.Create())

{

hashedDocument = sha256.ComputeHash(document); // hashing data bytes + putting in byte array

}

DigitalSignature.AssignNewKey(publicKeyPath, privateKeyPath);

var signature = cspParams.SignData(hashedDocument, privateKeyPath);

var verified = cspParams.VerifySignature(hashedDocument, signature, publicKeyPath);

Console.WriteLine(" Digital Signature Demonstration");

Console.WriteLine(" ---------------------------------------");

Console.WriteLine(" Original Text = " + Encoding.Default.GetString(document));

Console.WriteLine();

Console.WriteLine(" Digital Signature = " + Convert.ToBase64String(signature));

Console.WriteLine();

Console.WriteLine(verified

? " The digital signature has been correctly verified."

: " The digital signature has NOT been correctly verified.");

otherFunctions.prettyOutput();

}

} while (option != "0");

}

}

public class DigitalSignature

{

const string ContainerName = "MyContainer";

public static void AssignNewKey(string publicKeyPath, string privateKeyPath)

{

CspParameters cspParams = new CspParameters(1)

{

KeyContainerName = ContainerName,

Flags = CspProviderFlags.UseMachineKeyStore,

ProviderName = "Microsoft Strong Cryptographic Provider"

};

using (var rsa = new RSACryptoServiceProvider(2048)) // create a new instance of the RSACryptoServiceProvider

{

rsa.PersistKeyInCsp = true; // pass the previously created CspParameters object to its constructor

File.WriteAllText(privateKeyPath, rsa.ToXmlString(true));

File.WriteAllText(publicKeyPath, rsa.ToXmlString(false));

};

}

public byte[] SignData(byte[] hashOfDataToSign, string privateKeyPath)

{

using (var rsa = new RSACryptoServiceProvider())

{

rsa.PersistKeyInCsp = false;

rsa.FromXmlString(File.ReadAllText(privateKeyPath));

var rsaFormatter = new RSAPKCS1SignatureFormatter(rsa);

rsaFormatter.SetHashAlgorithm("SHA256");

return rsaFormatter.CreateSignature(hashOfDataToSign);

}

}

public bool VerifySignature(byte[] hashOfDataToSign, byte[] signature, string publicKeyPath)

{

using (var rsa = new RSACryptoServiceProvider())

{

rsa.FromXmlString(File.ReadAllText(publicKeyPath));

var rsaDeformatter = new RSAPKCS1SignatureDeformatter(rsa);

rsaDeformatter.SetHashAlgorithm("SHA256");

return rsaDeformatter.VerifySignature(hashOfDataToSign, signature);

}

}

}

public class otherFunctions

{

public static void prettyOutput() // for pretty output

{

Console.WriteLine();

Console.WriteLine(" | Press Enter to continue");

Console.ReadLine();

Console.WriteLine();

}

}

}

Виконання:

| Options:

| 1 - digital signature

| 0 - exit

Your option -> 1

Enter your data -> my dataaaa.

Digital Signature Demonstration

---------------------------------------

Original Text = my dataaaa.

Digital Signature = FI6nmLEzBE0oYWWkLe58ghNE0Qy11VvQxMn2yW6kR4TUIkr4/Yy6G8wznR6QxiB4OLXDiRR+f+Xr2Ydmx8oqKaCiv+2JLlDRjCpzQCcRttS9lY+zgzZIIf6XYpNMWgf0KnlROOyzUN7PcTbkdlHUQgRt3sIKTsUrizItOvGvMUKPrSxjFFUewMf75BHJTAL9Kzu2T4gHJ5l5C/LpmCsX4LUNBSMmRD8WQKAQG+kCCFv4gjZ4m1a08Dl2QspLWBaYsvDrpYGqGYiJRdMEUHKt4b5I/mTZDCSZPZUy3QuRsYJD8FATPGPkOQ11mRv87D0mEZAyC7sXFfXGAB7t4oBSRQ==

The digital signature has been correctly verified.

| Press Enter to continue

| Options:

| 1 - digital signature

| 0 - exit

Your option -> 0

D:\Lessons\2nd Year\Основи ?нформац?йної безпеки\Labs\Lab 8\lab8\lab8\bin\Debug\net6.0\lab8.exe (process 13848) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

Press any key to close this window . . .